LECTURE 8

LAST TIME

Circuit subclasses of P/poly

One-line summary: Even for seemingly weak classes, proving lower bounds appears ridiculously hard...

But proved superpoly lower bound for AC^0

LATEST NEWS

- Poll about future deadlines — please vote ASAP
- Pset 2 out. Deadline: see above.
- Peer evaluation of pset 1 due on Thursday

Questions ?

TODAY

Randomness as a computational resource

Lots of deep questions here — see Chap 8

We'll cut to the chase: study TMs that can flip fair random coins

DEF 1 Probabilistic Turing machine PTM

Turing machine with two transition functions δ_0, δ_1

In each step, apply δ_0 w prob 1/2 and δ_1 w prob 1/2.

$M(x)$ = output of M on x now random variable

PTM runs in $T(n)$-time if $\forall x$ halts in $\leq T(|x|)$ steps regardless of random choices

Similar to NDTM

- NDTM: accepts if $\exists$ one (out of exponentially many) accepting branch.
- PTM: look at fraction of accepting branches

For language $\mathcal{L} \subseteq \{0,1\}^*$ and $x \in \{0,1\}^*$, define

$$\mathcal{L}(x) = \begin{cases} 1 & \text{if } x \in \mathcal{L} \\ 0 & \text{if } x \notin \mathcal{L} \end{cases}$$

Our (main) model for efficient probabilistic/randomized computation:

$\boxed{BPP}$ bounded-error probabilistic polynomial time

<u>DEF 2</u> A PTM M decides $\mathcal{L}$ in time $T(n)$ if $\forall x$ M halts in $T(|x|)$ steps and $\Pr[M(x) = \mathcal{L}(x)] \geq 2/3$.

$BPTIME(T(n))$ = languages decided by PTMs in $O(T(n))$ time

$BPP = \bigcup_{c \in \mathbb{N}^+} BPTIME(n^c)$

Probability over <u>random choices</u>, not over input

Constant 2/3 arbitrary (will see later)

Don't need perfectly fair coins (but we'll ignore this)

<u>PROP 3</u> $\mathcal{L} \in BPP$ if exist poly-time (deterministic) TM M and polynomial p s.t. for every x

$$\Pr_{r \in_R \{0,1\}^{p(|x|)}} \left[M(x,r) = \mathcal{L}(x) \right] \geq 2/3$$

<u>Notational aside</u>

Uniform sampling from $\{0,1\}^n$: $x \in_R \{0,1\}^n$

$x \sim \{0,1\}^n$

$x \sim \mathcal{U}_n$

<u>COR 4</u> $P \subseteq BPP \subseteq EXP$

<u>Proof</u> Can try all possible random strings in exponential time and compute success probability

Can't prove even $BPP \neq NEXP$

What about BPP vs P?

Fairly strong reasons to believe $P = BPP$!

[Discussed in Chs 19-20 in Arora-Barak — we won't have time to cover this.]

Example of the power of randomization

POLYNOMIAL IDENTITY TESTING

Given: polynomial (multivariate) with integer coeff. In implicit form

Decide: Is the polynomial identically zero?

Representation <u>algebraic circuit</u>

Like Boolean circuits, but gates are $+, -, \times$

Can also have constants $0, 1, \ldots$ if we wish

Inputs $x_1, \ldots, x_n$

Single output node (sink)

Not hard to see: computes <u>some</u> polynomial

ZEROP = { algebraic circuits corresponding to polynomials that are identically zero }

Why <u>identity</u> testing?

Given C, C', construct $D = C - C'$ and check if $D \in$ ZEROP.

<u>Note</u> compact representation

$\prod_{i=1}^{n} (1 + x_i)$ has 2^n terms

circuit of size $2n$

SCHWARTZ- ZIPPEL LEMMA

Let $p(x_1, x_2, \dots, x_m)$ be non-zero poly of total degree $\leq d$. Let S finite set of integers. Then for $a_1, \dots, a_m$ chosen from S uniformly randomly with replacements

$$\Pr\left[p(a_1, \dots, a_m) \neq 0 \right] \geq 1 - \frac{d}{|S|}$$

<u>Proof</u> Induction over ~~m~~.

<u>Base case $m = 1$</u>: Univariate polynomial

Degree $\leq d \Rightarrow$ at most d roots

So p can evaluate to zero on at most d out of $|S|$ integers.

<u>Inductive step</u> See Arora-Barak App A.6

TESTING IDEA

Circuit of size $m \Rightarrow \leq m$ multiplications

$\Rightarrow$ degree $\leq 2^m$

So pick $a_1, \dots, a_m \in [1, 10 \cdot 2^m]$, evaluate circuit, and apply Schwartz-Zippel

If circuit C encodes zero poly $\Rightarrow$ result always 0

if non-zero poly $\Rightarrow$ 90% of non-zero output

<u>Problem</u> If degree $\approx 2^m$, then numbers grow as large as $(10 \cdot 2^m)^{2^m} \Rightarrow$ exponentially many bits.

Hard to do in poly time...

<u>Solution</u> "fingerprinting" compute modulo $k \in [2^{2m}]$

Computing modulo k
After each operation, divide by k
and take remainder

Suppose $y = C(a_1, \ldots, a_m)$

If $y = 0$, then $y = 0 \pmod{k}$

CLAIM 5

If $y \neq 0$, then randomly chosen $k \in [2^{2m}]$
will not divide y with prob $\geq \delta = \frac{1}{4m}$

Given this claim, run test $O(m)$ times
and accept only if always get 0 output

Proof of Claim 5

Assume $y \neq 0$. $y \leq (10 \cdot 2^m)^{2^m}$

Let $\mathcal{B}$ = prime factors of y.
Sufficient to show that with prob $\geq \delta$ k is
a prime not in $\mathcal{B}$

y has at most $\log y \leq 5m\, 2^m$ prime factors

By Prime Number Theorem — constant is actually 1 (*)

$$\#\text{primes} \leq N \sim N / \ln N$$

$$\#\text{primes} \leq 2^{2m} \sim \frac{2^{2m}}{2m} > \frac{2^{2m}}{4m} \quad \text{for large enough } m$$

$$5m\, 2^m = o\left(\frac{2^{2m}}{2m}\right) < \frac{2^{2m}}{8m} \quad \text{for large enough } m$$

$$\Pr[\,k \text{ prime not in } \mathcal{B}\,] \geq \frac{2^{2m}/8m}{2^{2m}} = \frac{1}{8m}$$

*) See Thm A.23 in Arora-Barak
for sufficient simple version

Many natural randomized algorithms have <u>one-sided error</u>

Might make mistake when $x \in L$ but never when $x \notin L$ or the other way round

(we just saw one such example)

<u>DEF 6</u> $RTIME(T(n))$ contains every language L for which $\exists$ PTM M running in time $O(T(n))$ such that

$$x \in L \Rightarrow \Pr[M(x) = 1] \geq 2/3$$
$$x \notin L \Rightarrow \Pr[M(x) = 0] = 1$$

$$RP = \bigcup_{c \in \mathbb{N}^+} RTIME(n^c)$$

<u>OBS 7</u> $RP \subseteq NP$

<u>Pf</u> Every accepting branch is a certificate.

Don't know if $BPP \subseteq NP$

RP: "Never false positives"

$coRP \doteq \{L \mid \bar{L} \in RP\}$ "Never false negatives"

Given general PTM M, can define random variable

$T_{M,x}$ = running time of M on x.

Take <u>expectation</u> of this random variable

$$E[T_{M,x}] = \sum_{t=1}^{\infty} t \cdot \Pr[T_{M,x} = t]$$

Say M has expected running time $T(n)$ if

$$\forall x \in \{0,1\}^* \quad E[T_{M,x}] \leq T(|x|)$$

DEF 8 $\mathrm{ZTIME}(T(n))$ contains all languages L for which $\exists$ PTM M that runs in expected time $O(T(n))$ such that

$$\Pr[\ M(x) = L(x) \mid M \text{ halts} \ ] = 1$$

$$ZPP = \bigcup_{c \in \mathbb{N}^+} \mathrm{ZTIME}(n^c)$$

"Zero-sided error"

ZPP zero-error probabilistic polynomial time

THM 9 $ZPP = RP \cap coRP$

Proof Exercise.

Also immediately clear from def

$$RP \subseteq BPP$$
$$coRP \subseteq BPP$$

ROBUSTNESS OF DEFINITIONS

(a) Error probability: constant 2/3 arbitrary

(b) Can use expected running time instead of worst case

(c) can use biased coins

(d) Can even use imperfect random sources ("weak random sources")

Will show (a) — see Sec 7.4 for the rest

LEMMA 10 For $c > 0$ constant, let $BPP_{1/2+n^{-c}}$ denote class of languages L for which $\exists$ poly-time PTM M s.t. $\forall x \in \{0,1\}^*$

$$\Pr[M(x) = L(x)] \geq \frac{1}{2} + |x|^{-c}$$

Then $BPP_{1/2+n^{-c}} = BPP$.

Need to show: can go from success prob $\frac{1}{2} + n^{-c}$ to $2/3$.

Show sth stronger: can go to $1 - 2^{-n^d}$

exponentially small failure prob.

THM 11 (ERROR REDUCTION FOR BPP)

Suppose $\exists$ poly-time PTM M for L s.t.

$$\forall x \quad \Pr[M(x) = L(x)] \geq 1/2 + |x|^{-c}.$$

Then $\forall d > 0$ $\exists$ poly-time PTM M' s.t.

$$\forall x \quad \Pr[M(x) = L(x)] \geq 1 - 2^{-|x|^d}$$

Proof

M' runs M for $k = 8|x|^{2c+d}$ times, collects answers, and takes majority vote.

How confident can we be that this is correct? Use material from App A.2.1 & A.2.4

Let $X_i = \begin{cases} 1 & \text{if } i\text{th run of } M \text{ gets } x \text{ right} \\ 0 & \text{otherwise} \end{cases}$

$\Pr[X_i = 1] = p$ for $p \geq \frac{1}{2} + |x|^{-c}$ [suppose $p = 1/2 + |x|^{-c}$ for simplicity]

$$E\left[\sum_{i=1}^{k} X_i\right] = kp = \underbrace{\frac{8|x|^{2c+d}}{2}}_{\text{half}} + \underbrace{8|x|^{c+d}}_{\text{margin}}$$

"If you repeat independent trials sufficiently many times, you will get very close to expected value with very high probability"

LEMMA 12 (CHERNOFF BOUND)

$$\Pr\Big[\,\big|\textstyle\sum_{i=1}^{k} X_i - pk\big| > \underbrace{\delta pk}_{\text{deviation from expected value}}\Big] < \exp\left(-\frac{\delta^2}{4}pk\right)$$

Plug in $p = \frac{1}{2} + |x|^{-c}$

$\delta = |x|^{-c}/2$

We will be correct unless $\sum_{i=1}^{k} X_i < pk - \delta pk$.
That probability is bounded by

$$\exp\left(-\frac{1}{4|x|^{2c}}\cdot\frac{8|x|^{2c+d}}{2}\right) = \exp\left(-|x|^d\right) < 2^{-|x|^d} \quad \boxed{}$$

Relationship between BPP and other classes?

<u>THM 12</u> $BPP \subseteq P/poly$

<u>THM 13</u> $BPP \subseteq \Sigma_2^P \cap \Pi_2^P \quad (\subseteq PH)$

Both proofs use error reduction in Thm 11 plus some other ideas.

Proof of Thm 13 is <u>extremely</u> neat...
But will have to skip it due to time constraints.

Try to sketch proof of Thm 12

Proof of Thm 12

If $L \in BPP$, then by Thm 11 (and Prop 3)
$\exists$ PTM M that on input size n

- uses m random bits
- gets answer right except with prob $2^{-(n+1)}$

Let r be the random bits
Say r <u>bad</u> for x if $M(x,r) \neq L(x)$

For every x, M succeeds with prob $\geq 1 - 2^{-(n+1)}$
$\Rightarrow$ out of 2^m random strings, $\leq 2^m / 2^{n+1}$ bad for x.

$$\left| \{ r \mid r \text{ bad for some } x \} \right| \leq \sum_{x \in \{0,1\}^n} \{ r \mid r \text{ bad for this } x \}$$

$$\leq \cancel{\tfrac{1}{2}}\, 2^n \cdot \frac{2^m}{2^{n+1}} = 2^m / 2$$

But this means that there is at least one random string $r^* \in \{0,1\}^m$ (in fact, at least half)

that are good for all $x \in \{0,1\}^n$

Run M with this r^* as advice!

Checking Thm 11 again, r^* will have poly size.

What about complete problems for BPP?

Typical complete problems
$\exists$ TM of correct type running with resource bound such-and-such

"Correct type": DTM - easy to check } syntactic
NDTM - easy to check }

BPP-style: accept x with prob $\geq 2/3$
or prob $\leq 1/3$
but not in between

Undecidable to check

Hierarchy theorems?

Fail for similar reasons.

A final useful notion: Randomized reductions

DEF 14 Language B reduces to language C under randomized reductions, denoted $B \leq_r C$, if $\exists$ PTM M s.t.

$$\forall x \in \{0,1\}^* \quad \Pr\left[C(M(x)) = B(x) \right] \geq 2/3$$

Not transitive
But if $C \in BPP$ and $B \leq_r C$ then $B \in BPP$

Could have defined NP in terms of randomized reductions instead (if BPP better formalization of "efficient computation)

$NP = \{ L \mid L \leq_p 3\text{-}SAT \}$

DEF 15

$$BP \cdot NP = \{ L \mid L \leq_r 3\text{-}SAT \}$$